Principal Investigator:	Andrew A. Chien
PI Institution:	University of Illinois at Urbana-Champaign
PI Phone Number:	217-333-6844
PI E-mail Address:	achien@cs.uiuc.edu
Grant or Contract Title:	Techniques for Efficient Execution of Fine-grained Massively-Parallel Programs
Grant or Contract Number:	N00014-93-1-1086 (AASERT linked with N00014-92-J-1961)
Final Report:	August 1997

Overview

The objective of the base project is to explore the use and implementation of object-oriented techniques for high performance, high-level programming. Object-oriented languages are a high level approach of clear programming benefit but are typically much less efficient that traditional approaches embodied in Fortran and C. We evaluate the complexity of programming irregular applications, and the efficient implementation of such programs using novel program analysis and runtime techniques.

We have developed a range of language implementation techniques which can significantly improve the efficiency of object-oriented programs. These techniques are embodied in the **Illinois Concert System**, and have been demonstrated on entire irregular, parallel application codes. The insights from our research are described in numerous technical publications, and were used to design a new parallel object-oriented language, ICC++.

Our research results have broad implications for the efficient implementation of both concurrent and sequential object-oriented languages. Major results include novel global type inference techniques [4] which produce detailed concrete type information as well as dynamic data structure analyses [5]. These techniques are embodied in the Illinois Concert System which is widely known and disseminated [1, 2]. Empirical evaluation of these techniques both on benchmark kernels and entire applications indicate they can allow object-oriented programs to match the sequential performance of C [6]. Our optimizations can produce performance equal to Fortran and message passing on large parallel applications, supporting a higher level programming model essential for programming complex pointer-based applications on scalable machines. In particular, we have demonstrated these techniques for protein molecular dynamics (IC-CEDAR) [7], parallel radiosity [3], and are working the Cosmology grand challenge team on an adaptive model of galactic formation. These results are documented in over thirty refereed publications in top conferences and journals.

1 Contributions of AASERT Students to Base Project

The AASERT participants have contributed significantly to the Concert project, both increasing the quality of the overall research effort and the breadth of study possible.

2 AASERT Students and their efforts

The students whose training and participation was enabled by the AASERT funding include John Plevyak, Kay Hane Connelly, Scott Pakin, Derek Taubert, and Brian Fin. Their efforts contributed to implementing and exploring additional compiler optimizations, evaluating and improving the performance of the Concert runtime system, and building irregular parallel applications which were used to evaluate the capabilities and performance of the Concert system. In particular, John Plevyak built a number of interprocedural analyses and compiler optimizations which form the core of the Concert System. Kay Connelly and Scott Pakin contributed to the runtime system evaluation and tuning. Derek and Brian contributed to the suite of irregular application programs used to evaluate the effectiveness of high level parallel programming techniques and their efficient implementation.

References

[1] Andrew Chien, Vijay Karamcheti, and John Plevyak. The Concert system—compiler and runtime support for efficient fine-grained concurrent object-oriented programs. Technical Report UIUCDCS-R-93-1815, Department of Computer Science, University of Illinois, Urbana, Illinois, June 1993.

[2] Andrew A. Chien and Julian Dolby. The Illinois Concert system: A problem-solving environment for irregular applications. In *Proceedings of DAGS'94, The Symposium on Parallel Computation and Problem Solving Environments.*, 1994.

[3] Vijay Karamcheti, John Plevyak, and Andrew A. Chien. Runtime mechanisms for efficient dynamic multithreading. *Journal of Parallel and Distributed Computing*, 37(1):21–40, 1996. Available from `http://www-csag.cs.uiuc.edu/papers/rtperf.ps`.

[4] John Plevyak and Andrew A. Chien. Precise concrete type inference of object-oriented programs. In *Proceedings of OOPSLA'94, Object-Oriented Programming Systems, Languages and Architectures*, pages 324–340, 1994.

[5] John Plevyak, Vijay Karamcheti, and Andrew Chien. Analysis of dynamic structures for efficient parallel execution. In *Proceedings of the Sixth Workshop for Languages and Compilers for Parallel Machines*, pages 37–56, August 1993.

[6] John Plevyak, Xingbin Zhang, and Andrew A. Chien. Obtaining sequential efficiency in concurrent object-oriented programs. In *Proceedings of the ACM Symposium on the Principles of Programming Languages*, pages 311–321, January 1995.

[7] Xingbin Zhang, Vijay Karamcheti, Tony Ng, and Andrew Chien. Optimizing COOP languages: Study of a protein dynamics program. In *IPPS'96*, 1996.